UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO E ENGENHARIA DE COMPUTAÇÃO

GRUPO DELTA
ÁTILA DA ROCHA COSTA E SILVA
BÉUREN FELIPE BECHLIN
FELIPE BERTOLDO COLOMBO DE SOUZA

Implementação do jogo War sando TypeScript

Relatório apresentado como requisito parcial para a obtenção de conceito na Disciplina de Modelos de Linguagens de Programação

Prof. Dr. Lucas Mello Schnorr
Orientador

Porto Alegre
2018

SUMÁRIO

1 INTRODUÇÃO

O trabalho visa a solução de um problema através de diferentes paradigmas de programação: orientação a objetos e funcional.

Com as duas implementações, teremos visto na prática as vantagens e desvantagens de cada paradigma na abordagem ao problema, tendo condições de estabelecer um comparativo entre diferentes pontos.

O problema escolhido foi o jogo War, versão brasileira do norte americano Risk, que será abordado a seguir.

1.1 Visão geral da linguagem

Dentro das linguagens disponibilizadas, TypeScript é a que possui maior suporte para elementos de interfaces e bibliotecas em geral, já que é um superset de JavaScript que, através do TypeScript Transpiler, é transformada para JavaScript.
Outro fator determinante é a necessidade de solucionar o problema também de uma forma funcional, e JavaScript é uma linguagem essencialmente funcional com vasta utilização de closures e padrões de projeto, como módulo.

Mesmo TypeScript sendo implementada para restringir o uso de JavaScript, além de adicionar tipagem estática e outros elementos de OOP, é possível utilizar esses padrões herdados da linguagem JavaScript, ajudando bastante em uma solução funcional.
Como já citado, o TypeScript foi desenvolvido com o intuito de criar uma linguagem mais estável para aplicações web e com melhor manutenibilidade que o JS. Adicionando elementos de checagem estática de tipos, diferentemente do JS, orientação a objetos com maior poder que as especificações de EcmaScript 6.

Dessa maneira, possuímos uma linguagem completa para solucionar um problema com grande interação com usuário e elementos gráficos como é um caso de um jogo.
Para exemplificar, é possível utilizar o framework de single page applications criado pelo Google chamado Angular, usado por grandes empresas. Também é possível utilizar uma biblioteca interface com usuário chamada React, desenvolvida e mantida pelo Facebook usada por Netflix, AirBnb e outros.

Para auxílio da programação funcional existe a implementação da biblioteca underscore para TypeScript. Além disso a linguagem possui um ótimo ambiente de desenvolvimento com vários utilitários desenvolvidos por terceiros para gerenciamento de

pacotes e dependências, gerenciadores de tarefas e frameworks para testes.

1.2 Descrição do problema

Em TypeScript será implementado o jogo War, versão brasileira do norteamericano Risk.

No jogo, o participante controla um exército que recebe um objetivo que deverá ser cumprido. O objetivo é conquistar determinado território. Se, ao final de uma batalha, o participante destruir todos os exércitos de defesa do adversário, ele terá conquistado o território.
Quando um objetivo é declarado como concluído, o participante que o declarou conquista o território de combate e recebe novos exércitos, aumentando seu poder.

Em cada rodada, o participante poderá receber novos exércitos, caso conquiste territórios, usá-los conforme sua estratégia, atacar outros exércitos ou mover exércitos.

Vence o participante que atingir o objetivo recebido no início do jogo.

2 DESENVOLVIMENTO

2.1 Dependências

2.2 Paradigma funcional

2.3 Paradigma de orientação a objetos

3 RESULTADOS

4 CONCLUSÃO

REFERÊNCIAS

The TypeScript Reference,
https://www.typescriptlang.org/docs/home.html

The ReactJS Reference,
https://reactjs.org/docs/react-api.html

Using Redux with React,
https://redux.js.org/basics/usage-with-react

Underscore JS,
http://underscorejs.org/

React and Redux TypeScript Guide,
https://github.com/piotrwitek/react-redux-typescript-guide